

(12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum  
Internationales Büro



(43) Internationales Veröffentlichungsdatum  
23. Oktober 2003 (23.10.2003)

PCT

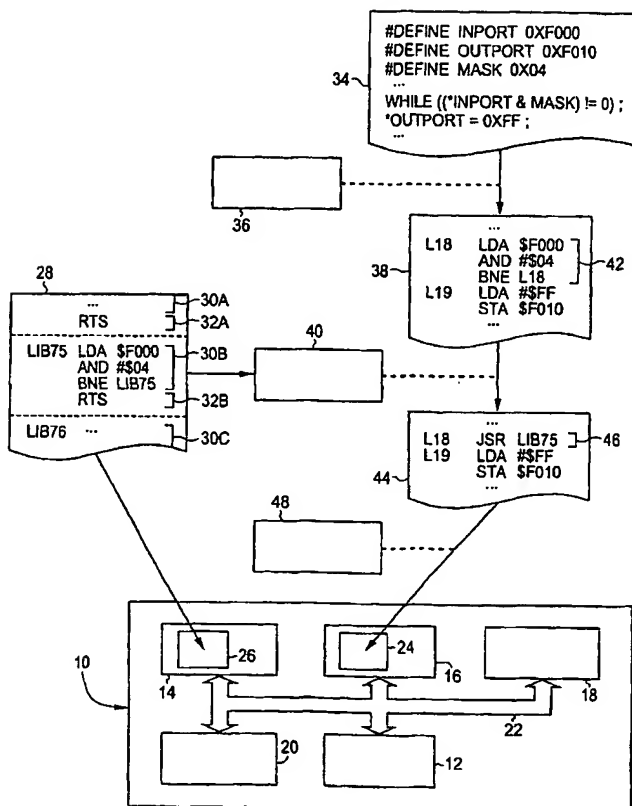
(10) Internationale Veröffentlichungsnummer  
WO 03/088039 A2

- (51) Internationale Patentklassifikation<sup>7</sup>: G06F 9/45 (71) Anmelder (für alle Bestimmungsstaaten mit Ausnahme von US): GIESECKE & DEVRIENT GMBH [DE/DE]; Prinzregentenstrasse 159, 81677 München (DE).
- (21) Internationales Aktenzeichen: PCT/EP03/03803 (72) Erfinder; und (75) Erfinder/Anmelder (nur für US): BALDISCHWEILER, Michael [DE/DE]; Hans-Jakob-Str. 99, 81825 München (DE). NESS, Werner [DE/DE]; Ringhofferstr. 76, 85716 Unterschleißheim (DE).
- (22) Internationales Anmeldedatum: 11. April 2003 (11.04.2003) (74) Anwalt: DENDORFER, Claus; Wächtershäuser & Hartz, European Patent and Trademark Attorneys, Weinstrasse 8, 80333 München (DE).
- (25) Einreichungssprache: Deutsch
- (26) Veröffentlichungssprache: Deutsch
- (30) Angaben zur Priorität: 102 16 602.1 15. April 2002 (15.04.2002) DE

[Fortsetzung auf der nächsten Seite]

(54) Title: OPTIMISATION OF A COMPILER GENERATED PROGRAM CODE

(54) Bezeichnung: OPTIMIERUNG VON COMPILERGENERIERTEM PROGRAMMCODE



(57) Abstract: The invention relates to a method for the optimisation of a compiler generated program code (38). The program code (38) generated by the compiler is examined by program code fragments (42) which at least correspond in relation to the action thereof to a library-code fragment (30x) which is contained in a predefined library (28). The thus discovered program-code-fragments (42) are replaced by respectively calling up the corresponding library-code fragment (30x). A computer program product comprises program instructions for carrying out said method. A mobile data carrier (10) contains the program code (44), which is optimised according to said method, and containing a library. The invention enables efficient utilisation of the memory available in the mobile data carrier (10) and offers a high degree of flexibility for program modifications and/or for the production of data carriers (10) in relatively small quantities.

(57) Zusammenfassung: Bei einem Verfahren zur Optimierung von compilergeneriertem Programmcodem wird der compilergenerierte Programmcodem 38 nach Programm-Codefragmenten 42 durchsucht, die zumindest hinsichtlich ihrer Wirkung je einem in einer vordefinierten Bibliothek 28 enthaltenen Bibliotheks-Codefragment 30x entsprechen. Die dabei aufgefundenen Programm-Code-fragmente 42 werden durch je einen Aufruf des entsprechenden Bibliotheks-Codefragments 30x ersetzt. Ein Computerprogrammprodukt weist Programminstruktionen zur Ausführung dieses Verfahrens auf. Ein tragbarer Datenträger 10 enthält den gemäss diesem Verfahren optimierten Programmcodem 44 sowie die Bibliothek 28. Die Erfindung ermöglicht eine gute Ausnutzung des bei tragbaren Datenträgern 10

vorhandenen Speichers und eine hohe Flexibilität

[Fortsetzung auf der nächsten Seite]



(81) **Bestimmungsstaaten (national):** AE, AG, AL, AM, AT (Gebrauchsmuster), AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ (Gebrauchsmuster), CZ, DK (Gebrauchsmuster), DK, DM, DZ, EC, EE (Gebrauchsmuster), EE, ES, FI (Gebrauchsmuster), FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK (Gebrauchsmuster), SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) **Bestimmungsstaaten (regional):** ARIPO-Patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), eurasisches Patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), europäisches Patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI-Patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

**Erklärung gemäß Regel 4.17:**

- hinsichtlich der Berechtigung des Anmelders, ein Patent zu beantragen und zu erhalten (Regel 4.17 Ziffer ii) für die folgenden Bestimmungsstaaten AE, AG, AL, AM, AT, AU,

AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW, ARIPO-Patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), eurasisches Patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), europäisches Patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI-Patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG)

**Veröffentlicht:**

- ohne internationalen Recherchenbericht und erneut zu veröffentlichen nach Erhalt des Berichts

Zur Erklärung der Zweibuchstaben-Codes und der anderen Abkürzungen wird auf die Erklärungen ("Guidance Notes on Codes and Abbreviations") am Anfang jeder regulären Ausgabe der PCT-Gazette verwiesen.

### Optimierung von compilergeneriertem Programmcode

Die Erfindung betrifft die Programmierung von tragbaren Datenträgern sowie die Programmausführung durch tragbare Datenträger. Ein tragbarer Datenträger im Sinne des vorliegenden Dokuments kann insbesondere eine Chipkarte (*smart card*) in unterschiedlichen Bauformen oder ein Chipmodul sein.

Tragbare Datenträger, wie sie gegenwärtig üblich sind, weisen einen Prozessorkern und mehrere in unterschiedlichen Technologien gefertigte Speicher auf. In einer typischen Konfiguration sind beispielsweise ein maskenprogrammiertes ROM, ein elektrisch lösch- und programmierbares EEPROM und ein beschreibbares RAM vorgesehen. Das RAM dient als Arbeitsspeicher während des Programmblaufs, während der vom Prozessorkern auszuführende Programmcode im ROM und/oder im EEPROM abgelegt sein kann. Diese und ähnliche Ausgestaltungen von tragbaren Datenträgern sind in Abschnitt 3.4 des Buches "Handbuch der Chipkarten" von W. Rankl und W. Effing, Hanser Verlag, 3. Auflage 1999, beschrieben.

Typischerweise belegt eine Speicherzelle im EEPROM ungefähr die vierfache Chipfläche wie eine ROM-Speicherzelle. Aus Gründen der Flächeneinsparnis bzw. der höheren verfügbaren Speicherkapazität bei gleicher Fläche ist es daher wünschenswert, den ausführbaren Programmcode möglichst weitgehend im ROM unterzubringen. Allerdings muß der Inhalt des maskenprogrammierten ROM schon bei der Herstellung für große Stückzahlen des Datenträgers unveränderlich festgelegt werden. Das EEPROM wird dagegen erst beim Komplettieren und Initialisieren einer Serie von Datenträgern bzw. beim Personalisieren der einzelnen Datenträger beschrieben. Eine möglichst weitgehende Speicherung des ausführbaren Programmcodes im EEPROM ist daher wegen der höheren Flexibilität vorteilhaft. Dies betrifft sowohl die

- 2 -

Programmierung kleinerer Stückzahlen von Datenträgern als auch die Fehlerkorrektur und das Einbringen zusätzlicher Funktionen bei Großserien.

5 Es besteht daher das Problem, bei der Programmierung von tragbaren Datenträgern einerseits das maskenprogrammierte ROM oder einen vergleichbaren Speicher möglichst weitgehend zu nutzen und andererseits eine möglichst hohe Flexibilität für Programmänderungen und/oder für die Herstellung von Datenträgern in kleineren Stückzahlen zu erzielen.

10 Erfindungsgemäß wird dieses Problem ganz oder zum Teil gelöst durch ein Verfahren mit den Merkmalen des Anspruchs 1, ein Computerprogrammprodukt gemäß Anspruch 8 und einen tragbaren Datenträger gemäß Anspruch 10. Die abhängigen Ansprüche definieren bevorzugte Ausgestaltungen der Erfindung. Die Aufzählungsreihenfolge der Schritte in den Ver-  
15 fahrensansprüchen soll nicht als Einschränkung des Schutzbereichs verstanden werden. Es sind vielmehr Ausgestaltungen der Erfindung vorgesehen, in denen diese Schritte in anderer Reihenfolge oder ganz oder teilweise parallel oder ganz oder teilweise ineinander verzahnt ausgeführt werden.

20 Die Erfindung geht von der Grundidee aus, zur Optimierung des Programmcodes eine vordefinierte Bibliothek einzusetzen, die eine Mehrzahl von Bibliotheks-Codefragmenten enthält. In dem erfindungsgemäßen Optimierungsvorgang wird der zu optimierende Programmcode seinerseits nach Programm-Codefragmenten durchsucht, die in ihrer Wirkung oder Funktion  
25 je einem Bibliotheks-Codefragment entsprechen. Solche Programm-Codefragmente werden durch je einen Aufruf des entsprechenden Bibliotheks-Codefragments ersetzt. Der optimierte Programmcode wird in einem ersten Speicherbereich des Datenträgers (z.B. im EEPROM) abgelegt, während die

Bibliothek zur Speicherung in einem zweiten Speicherbereich (z.B. im ROM) vorgesehen ist.

5 Der erfindungsgemäße Optimierungsvorgang führte bei von den Erfindern vorgenommenen Tests zu einer deutlichen Reduktion der Größe des für den ersten Speicherbereich vorgesehenen Programmcodes. Dieses Ergebnis ist überraschend, weil man intuitiv annehmen würde, daß sich bei realistischem Umfang der Bibliothek nur wenige Übereinstimmungen von Teilen des Programmcodes mit den Bibliotheks-Codefragmenten ergeben würden.

10 Die durch die Erfindung bewirkte Verringerung der Codegröße hat zur Folge, daß bei einem Datenträger mit vorgegebener Speicherausstattung Programmcodes für zusätzliche Funktionen in den ersten Speicherbereich aufgenommen werden kann. Ist der erste Speicherbereich als EEPROM oder  
15 in vergleichbarer Technologie ausgestaltet, so braucht dieser Programmcodes erst beim Kompletieren oder Initialisieren oder Personalisieren des Datenträgers geladen zu werden. Eine Änderung oder Neuerstellung des Programmcodes, der wegen seiner Kompaktheit eine Vielzahl von Funktionen implementiert, ist daher erstens schnell und zweitens schon für kleine Stück-  
20 zahlen von Datenträgern oder sogar für einzelne Datenträger möglich.

Die vordefinierte Bibliothek befindet sich erfindungsgemäß im zweiten Speicherbereich, also z.B. im maskenprogrammierten ROM. In der Regel ist die durch die erfindungsgemäße Optimierung erzielte Einsparung von Pro-  
25 grammcode geringer als die Bibliotheksgröße. Auch in diesem Fall ist jedoch der Einsatz der Erfindung vorteilhaft, weil der wertvolle erste Speicherbereich besser genutzt wird. Falls sich im compilergenerierten Programmcodes viele Codefragmente befinden, die jeweils gruppenweise durch je ein einziges Codefragment der Bibliothek ersetzt werden können, und falls die

Bibliothek nur wenige nicht benötigte Codefragmente enthält, kann der Programmcode durch die Optimierung sogar um mehr als die Bibliothekslänge schrumpfen. In diesem Fall ist die Verwendung der Erfindung sogar dann vorteilhaft, wenn der erste und der zweite Speicherbereich nur konzeptuelle Abschnitte ein und desselben physischen Speicherfeldes sind.

Erfindungsgemäß wird zur Optimierung nach Programm-Codefragmenten gesucht, also nach Abschnitten im compilergenerierten Programmcode, die sich durch entsprechende Bibliotheks-Codefragmente ersetzen lassen. Bei der Programmerstellung braucht der Programmierer keine Rücksicht auf diesen späteren Optimierungsvorgang zu nehmen; insbesondere braucht er im Programm keine Aufrufe von Bibliotheksroutinen vorzusehen. Die Programmentwicklung wird daher durch die Erfindung in keiner Weise erschwert.

In der hier verwendeten Wortwahl sollen die Begriffe "Programmcode" oder "Codefragment" sowohl ausführbaren Maschinencode vor oder nach dem Binden als auch den entsprechenden Assembler-Quellcode bezeichnen. Mit anderen Worten kann in unterschiedlichen Ausgestaltungen der Erfindung der erfindungsgemäße Optimierungsvorgang sowohl auf Grundlage des compilergenerierten Assembler-Quellcodes als auch auf Grundlage des bereits assemblierten Maschinencodes vorgenommen werden. Im erstgenannten Fall erfolgen die Assemblierung und gegebenenfalls das Binden erst nach der Optimierung. Die Bibliothek kann während der Optimierung ebenfalls als Assembler-Quellcode und/oder als bereits assemblierter Maschinencode vorliegen.

Generell ist eine Ersetzung eines Programm-Codefragments durch ein Bibliotheks-Codefragment immer dann möglich, wenn beide Codefragmente

einander entsprechende Funktionen ausführen. Hier können komplexe Berechnungen hinsichtlich der genauen Wirkungen von Codefragmenten vorgenommen werden, um z.B. auch dann einen Ersetzungsvorgang auszulösen, wenn einzelne Instruktionen in den Codefragmenten in einer un-

5 schädlichen Weise vertauscht sind. In besonders einfachen Ausführungsbeispielen wird dagegen nur dann eine Ersetzung vorgenommen, wenn die Codefragmente hinsichtlich des durch sie definierten Maschinencodes identisch sind. Auch bei dieser einfachen Ausgestaltung ist jedoch eine gewisse Analyse der Codefragmente erforderlich, weil z.B. ein Codefragment,

10 das einen Sprung mit einem nicht im Codefragment liegenden Sprungziel aufweist, in der Regel nicht ersetzt werden darf.

Zusätzliche Ersetzungsmöglichkeiten ergeben sich, wenn parametrisierte Codefragmente verwendet werden, die ähnlich einem Prozeduraufruf einen

15 oder mehrere Parameter (z.B. Speicheradressen oder Zahlenwerte) enthalten.

Vorzugsweise erfolgt der Aufruf eines Bibliotheks-Codefragments in der Regel durch einen in den Programmcode eingefügten Unterprogramm-Aufrufbefehl. In der Bibliothek ist dann ein unmittelbar auf das Bibliotheks-

20 Codefragment folgender Rücksprungbefehl vorgesehen. Ausnahmen von dieser Regel können in manchen Ausführungsformen dann gelten, wenn das zu ersetzende Codefragment in den Programmfluß eingreift. Falls z.B. das Codefragment mit einem Unterprogramm-Rücksprungbefehl endet, so kann der Aufruf in der Regel mittels eines Sprungbefehls erfolgen.

25

Die verwendete Bibliothek ist erfindungsgemäß vordefiniert, also nicht von dem im aktuellen Optimierungslauf verarbeiteten Programmcode abhängig. Um möglichst gute Optimierungsergebnisse zu erzielen, ist die Bibliothek jedoch vorzugsweise so gestaltet, daß sie geeignete Einträge für häufig vor-

kommende Strukturen des Programmcodes enthält. Solche häufig auftretenden Codeabschnitte können insbesondere von der Hardware und/oder einem Betriebssystem des Datenträgers und/oder von einem bei der Erzeugung des compilergenerierten Programmcodes eingesetzten Compiler abhängen.

Das erfindungsgemäß vorgesehene Computerprogrammprodukt kann insbesondere ein computerlesbarer Datenträger wie z.B. ein elektronisches oder magnetisches oder optisches Speichermedium sein, es ist aber nicht auf körperliche Datenträger beschränkt. Auch elektrische oder optische Signale (z.B. Spannungspegel einer Kommunikationsverbindung) sollen im hier verwendeten Sinne als Computerprogrammprodukt aufgefaßt werden. Das Computerprogrammprodukt enthält Programmcode, der die erfindungsgemäßen Optimierungsschritte ausführt. Vorzugsweise enthält das Computerprogrammprodukt ferner einen Compiler und/oder einen Assembler und/oder ein Bindeprogramm und/oder ein Ladeprogramm.

Das erfindungsgemäße Computerprogrammprodukt und der erfindungsgemäße tragbare Datenträger sind bevorzugt mit Merkmalen weitergebildet, die den oben beschriebenen und/oder in den Verfahrensansprüchen genannten Merkmalen entsprechen.

Weitere Merkmale, Aufgaben und Vorteile der Erfindung gehen aus der folgenden Beschreibung eines Ausführungsbeispiels und mehrerer Ausführungsalternativen hervor. Es wird auf die schematische Zeichnung verwiesen, in der die einzige Figur (Fig. 1) eine Darstellung eines tragbaren Datenträgers sowie unterschiedlicher Fassungen des Programmcodes in einem Ausführungsbeispiel der Erfindung zeigt.



- 7 -

Die Erfindung wird bei der Programmierung eines tragbaren Datenträgers 10 eingesetzt, der im hier beschriebenen Ausführungsbeispiel als Chipkarte ausgestaltet ist. In an sich bekannter Weise enthält der Datenträger 10 einen Halbleiterchip mit einem Prozessorkern 12, einem maskenprogrammierten ROM 14, einem EEPROM 16, einem RAM 18 und einer Schnittstelle 20 zur kontaktlosen oder kontaktgebundenen Kommunikation. Die genannten Komponenten sind über einen Bus 22 miteinander verbunden. In Ausführungsalternativen können die drei Speicherfelder 14, 16, 18 in anderen Technologien ausgestaltet sein; insbesondere kann die Flash-Technologie für das ROM 14 und/oder das EEPROM 16 eingesetzt werden.

In den Speicherfeldern 14, 16, 18 sind konzeptuell ein erster und ein zweiter Speicherbereich 24, 26 vorgesehen. Der erste Speicherbereich 24 dient zur Aufnahme des optimierten Programmcodes in Form von ausführbarem Maschinencode. Im zweiten Speicherbereich 26 wird eine vordefinierte Bibliothek 28 ebenfalls in Form von ausführbarem Maschinencode abgelegt. Der erste Speicherbereich 24 befindet sich im hier beschriebenen Ausführungsbeispiel im EEPROM 16, und der zweite Speicherbereich 26 befindet sich im ROM 14. In an sich bekannter Weise enthält das ROM 14 außer dem zweiten Speicherbereich 26 weitere, fest vorgegebene Routinen, die z.B. ein Betriebssystem des Datenträgers 10 bilden. Das EEPROM 16 enthält ferner ein Dateisystem für Daten, die nicht-flüchtig im Datenträger 10 gespeichert werden sollen.

Die Bibliothek 28 weist eine Vielzahl von vordefinierten Bibliotheks-Codefragmenten 30A, 30B, 30C, ... auf, die im folgenden allgemein mit 30x bezeichnet werden. In Fig. 1 sind die Bibliotheks-Codefragmente 30x aus Gründen der klareren Darstellung als Assembler-Sourcecode gezeigt. In der Regel folgt auf jedes Bibliotheks-Codefragment 30x unmittelbar ein Unter-

programm-Rücksprungbefehl 32A, 32B, ... (im folgenden allgemein mit 32x bezeichnet). Der Unterprogramm-Rücksprungbefehl 32x kann jedoch entfallen, wenn er bei der Ausführung des Bibliotheks-Codefragments 30x nicht erreicht werden kann, weil beispielsweise jeder Programmablauf des Bibliotheks-Codefragments 30x in einem Aussprung oder in einem im Bibliotheks-Codefragment 30x enthaltenen Unterprogramm-Rücksprung endet.

Die Programmentwicklung für den tragbaren Datenträger 10 geht von einem hochsprachlichen Quellcode 34 aus, der in Fig. 1 beispielhaft in der Programmiersprache C dargestellt ist. Der in Fig. 1 gezeigte Ausschnitt wartet, bis das von der Einerstelle aus dritte Bit des Eingaberegisters INPORT den Wert "0" einnimmt, und setzt dann das Ausgaberegister OUTPORT auf den Hexadezimalwert "FF". Ein an sich bekannter Compiler 36 setzt den hochsprachlichen Quellcode 34 in compilergenerierten Programmcode 38 um, der in Fig. 1 in Form von Assembler-Quellcode für den 6805-Befehlssatz dargestellt ist. In Ausführungsalternativen sind andere Befehlssätze, jeweils entsprechend dem Prozessorkern 12, vorgesehen.

Ein Optimierungsprogramm 40 führt die für das vorliegende Ausführungsbeispiel wesentlichen Optimierungsschritte aus. Das Optimierungsprogramm 40 verarbeitet den compilergenerierten Programmcode 38 und greift überdies auf Informationen über die in der Bibliothek 28 enthaltenen Bibliotheks-Codefragmente 30x zu. In unterschiedlichen Ausführungsvarianten kann in diesen Informationen z.B. eine Kopie der Bibliothek 28 im Assembler-Quellcode und/oder eine Kopie der Bibliothek 28 im ausführbaren Maschinencode und/oder eine Spezifikation der Wirkung der einzelnen Bibliotheks-Codefragmente 30x in einer geeigneten Beschreibungssprache enthalten sein. Es können ferner Zusatzinformationen wie z.B. Indizes

oder Hash-Tabellen vorgesehen sein, um die vom Optimierungsprogramm 40 vorgenommenen Suchvorgänge zu beschleunigen.

Das Optimierungsprogramm 40 identifiziert im compilergenerierten Programmcode 38 enthaltene Programm-Codefragmente 42, die bei der Ausführung durch den Prozessorkern 12 eine identische Funktion aufweisen wie in der Bibliothek 28 enthaltene Bibliotheks-Codefragmente 30x. Im vorliegenden Ausführungsbeispiel wird dazu ein relativ einfaches Verfahren eingesetzt, bei dem der compilergenerierte Programmcode 38 auf Assembler-  
10 Quelltextebene mit den einzelnen Einträgen in der Bibliothek 28 verglichen wird. Hinsichtlich der Befehlskürzel und der Adressen- und Werteangaben kann dabei ein textueller Vergleich stattfinden. Symbolische Sprungziele müssen dagegen vor dem Vergleich in eine standardisierte Form oder in einen numerischen Relativwert umgewandelt werden. In Ausführungsalternativen kann die Optimierung dagegen auf Grundlage eines compilergenerierten Programmcodes 38 erfolgen, der bereits in Form von assemblier-  
15 tem Maschinencode vorliegt.

Ein Programm-Codefragment 42, für das beim Vergleichsvorgang ein entsprechendes Bibliotheks-Codefragment 30x gefunden wurde, wird bei dem  
20 Optimierungsvorgang durch einen Aufruf dieses Bibliotheks-Codefragments 30x ersetzt. In Fig. 1 sind beispielsweise das Programm-Codefragment 42 und das Bibliotheks-Codefragment 30B bis auf die symbolische Benennung des Sprungziels identisch. Daher ersetzt das Optimierungsprogramm 40  
25 dieses Programm-Codefragment 42 im optimierten Programmcode 44 durch einen Aufruf des Bibliotheks-Codefragments 30B. Im vorliegenden Beispiel ist dieser Aufruf als Unterprogramm-Aufrufbefehl 46 ausgestaltet. Da das Programm-Codefragment 42 im vorliegenden Beispiel einem Maschinencode von sieben Bytes Länge entspricht und der Unterprogramm-Aufrufbefehl 46

- 10 -

nur drei Bytes benötigt, wurde der für den optimierten Programmcode 44 benötigte Speicherplatz durch die Ersetzung beträchtlich verringert.

Nach Abschluß der Optimierung wird der optimierte Programmcode 44  
5 durch einen Assembler 48 in vom Prozessorkern 12 ausführbaren Maschinencode umgesetzt. Nach einem gegebenenfalls erforderlichen Bindevorgang mit weiteren Programmteilen wird der Code beim Kompletieren oder Initialisieren oder Personalisieren des Datenträgers 10 in den ersten Speicherbereich 24 geladen. Die Bibliothek 28 befindet sich bereits seit der Chip-  
10 herstellung des Datenträgers 10 im zweiten Speicherbereich 26. Der Datenträger 10 ist damit einsatzbereit. Die oben beschriebenen Übersetzungs-, Optimierungs- und Assemblierungsschritte werden von einem Allzweckcomputer (in Fig. 1 nicht gezeigt) vorgenommen, der den Compiler 36, das Optimierungsprogramm 40 und den Assembler 48 ausführt.

15 Wenn im Betrieb des Datenträgers 10 die Programmausführung durch den Prozessorkern 12 im ersten Speicherbereich 24 an die Stelle des Unterprogramm-Aufrufbefehls 46 gelangt, wird das Bibliotheks-Codefragment 30B im zweiten Speicherbereich 26 als Unterprogramm ausgeführt. In ihrer Wirkung  
20 entsprechen die ausgeführten Instruktionen genau dem bei der Optimierung entfernten Programm-Codefragment 42. Nach der Ausführung dieser Instruktionen führt der Prozessorkern 12 einen durch den Unterprogramm-Rücksprungbefehl 32B ausgelösten Rücksprung an denjenigen Befehl im ersten Speicherbereich 24 aus, der unmittelbar auf den Unterprogramm-  
25 Aufrufbefehl 46 folgt.

Bei der Optimierung ist darauf zu achten, daß die Programmfunktionen nicht verändert werden. So sollten beispielsweise Programm-Codefragmente 42 mit Sprungbefehlen, die möglicherweise ein außerhalb des Programm-

Codefragments 42 liegendes Sprungziel aufweisen, nur nach genauer Analyse ersetzt werden. Eine Ersetzung ist zulässig, wenn jeder mögliche Ablauf des Programm-Codefragments 42 mit einem Aussprung oder einem Unterprogramm-Rücksprung endet. In diesen Fällen erfolgt der Aufruf des  
5 entsprechenden Bibliotheks-Codefragments 30x jedoch nicht mit einem Unterprogramm-Aufrufbefehl, sondern mit einem normalen Sprungbefehl. Diese Überlegungen können auch schon bei der Erstellung der Bibliothek 28 einfließen, so daß diese nur solche Bibliotheks-Codefragmente 30x enthält, deren Verwendung ohne weitere Nebenbedingungen zulässig ist.

10

Die Bibliothek 28 sollte so aufgebaut werden, daß sie möglichst oft geeignete Bibliotheks-Codefragmente 30x bereitstellt und somit möglichst viele Optimierungsmöglichkeiten bietet. So ist das Bibliotheks-Codefragment 30B von Fig. 1 beispielsweise auf die Hardware-Eigenschaften des Datenträgers 10  
15 abgestimmt. Wenn das in diesem Bibliotheks-Codefragment 30B abgefragte Eingabebit einem häufig benötigten Signalwert entspricht, so ist davon auszugehen, daß sich entsprechende Programm-Codefragmente 42 immer wieder im compilergenerierten Programmcode 42 für die unterschiedlichsten Anwendungen des Datenträgers 10 finden. Auf ähnliche Weise können häufige Betriebssystem-Aufrufe durch entsprechende Bibliotheks-Codefrag-  
20 mente 30x abgedeckt werden. Eine weitere Quelle für sich wiederholende Codefragmente im compilergenerierten Programmcode 38 ergibt sich durch die Tatsache, daß die Codeerzeugung im Compiler 36 schematisch abläuft und daher wiederkehrende Codestrukturen generiert werden.

25

Insgesamt ist es daher vorteilhaft, zur Erzeugung der Bibliothek 28 den vom Compiler 36 erzeugten Programmcode 38 für eine Vielzahl von Anwendungen, die für die Hardware und das Betriebssystem des Datenträgers 10 vorgesehen sind, statistisch auszuwerten.

Patentansprüche

1. Verfahren zur Optimierung von compilergeneriertem Programmcode (38), der für einen tragbaren Datenträger (10) mit einem Prozessorkern (12) sowie einem ersten und einem zweiten Speicherbereich (24, 26) vorgesehen ist, wobei:
- der erste Speicherbereich (24) zur Aufnahme des optimierten Programmcodes (44) vorgesehen ist,
  - der zweite Speicherbereich (26) zur Aufnahme einer vordefinierten Bibliothek (28) mit einer Mehrzahl von Bibliotheks-Codefragmenten (30x) vorgesehen ist, und
  - der compilergenerierte Programmcode (38) nach Programm-Codefragmenten (42) durchsucht wird, die zumindest hinsichtlich ihrer Wirkung je einem Bibliotheks-Codefragment (30x) entsprechen, wobei die dabei aufgefundenen Programm-Codefragmente (42) durch je einen Aufruf des entsprechenden Bibliotheks-Codefragments (30x) ersetzt werden.
2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß ein Programm-Codefragment (42) nur dann durch ein Bibliotheks-Codefragment (30x) ersetzt wird, wenn beide Codefragmente (42, 30x) in ihrer Form als ausführbarer Maschinencode identisch sind.
3. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß zumindest einige Bibliotheks-Codefragmente (30x) parametrisiert sind.
4. Verfahren nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, daß ein zu ersetzendes Programm-Codefragment (42)

- 13 -

zumindest dann, wenn es nicht in den Programmfluß eingreift, durch einen Unterprogramm-Aufrufbefehl (46) zu dem entsprechenden Bibliotheks-Codefragment (30x) ersetzt wird.

- 5           5. Verfahren nach einem der Ansprüche 1 bis 4, dadurch gekennzeichnet, daß der compilergenerierte Programmcode (38) in Form von Assembler-Quellcode vorliegt, und daß der Optimierungsvorgang auf Quellcode-Ebene durchgeführt wird.
- 10          6. Verfahren nach einem der Ansprüche 1 bis 5, dadurch gekennzeichnet, daß die vordefinierte Bibliothek (28) an die Hardware des tragbaren Datenträgers (10) und/oder an ein Betriebssystem des tragbaren Datenträgers (10) und/oder an einen bei der Erzeugung des compilergenerierten Programmcodes (38) eingesetzten  
15          Compiler (36) angepaßt ist.
- 20          7. Verfahren nach einem der Ansprüche 1 bis 6, dadurch gekennzeichnet, daß der erste Speicherbereich (24) elektrisch programmierbar ist, und/oder daß der zweite Speicherbereich (26) maskenprogrammierbar ist, und/oder daß der erste Speicherbereich (24) im tragbaren Datenträger (10) mehr Chipfläche pro Speicherzelle beansprucht als der zweite Speicherbereich (26).
- 25          8. Computerprogrammprodukt mit Programminstruktionen für einen Allzweckrechner, die den Allzweckrechner dazu veranlassen, ein Verfahren nach einem der Ansprüche 1 bis 7 auszuführen.
9. Computerprogrammprodukt nach Anspruch 8, dadurch gekennzeichnet, daß die Programminstruktionen ferner einen Compiler

- 14 -

(36) zum Umsetzen eines hochsprachlichen Quellcodes (34) in den compilergenerierten Programmcode (38) implementieren.

- 5           10. Tragbarer Datenträger (10) mit einem Prozessorkern (12), einem  
ersten Speicherbereich (24) und einem zweiten Speicherbereich  
            (26), wobei in dem ersten Speicherbereich (24) optimierter Pro-  
            grammcode (44) enthalten ist, der durch ein Verfahren nach einem  
            der Ansprüche 1 bis 7 erzeugt wurde, und in dem zweiten Spei-  
            cherbereich (26) eine unabhängig vom optimierten Programm-  
10           code (44) vordefinierte Bibliothek (28) mit einer Mehrzahl von  
            Bibliotheks-Codefragmenten (30x) enthalten ist.



1/1

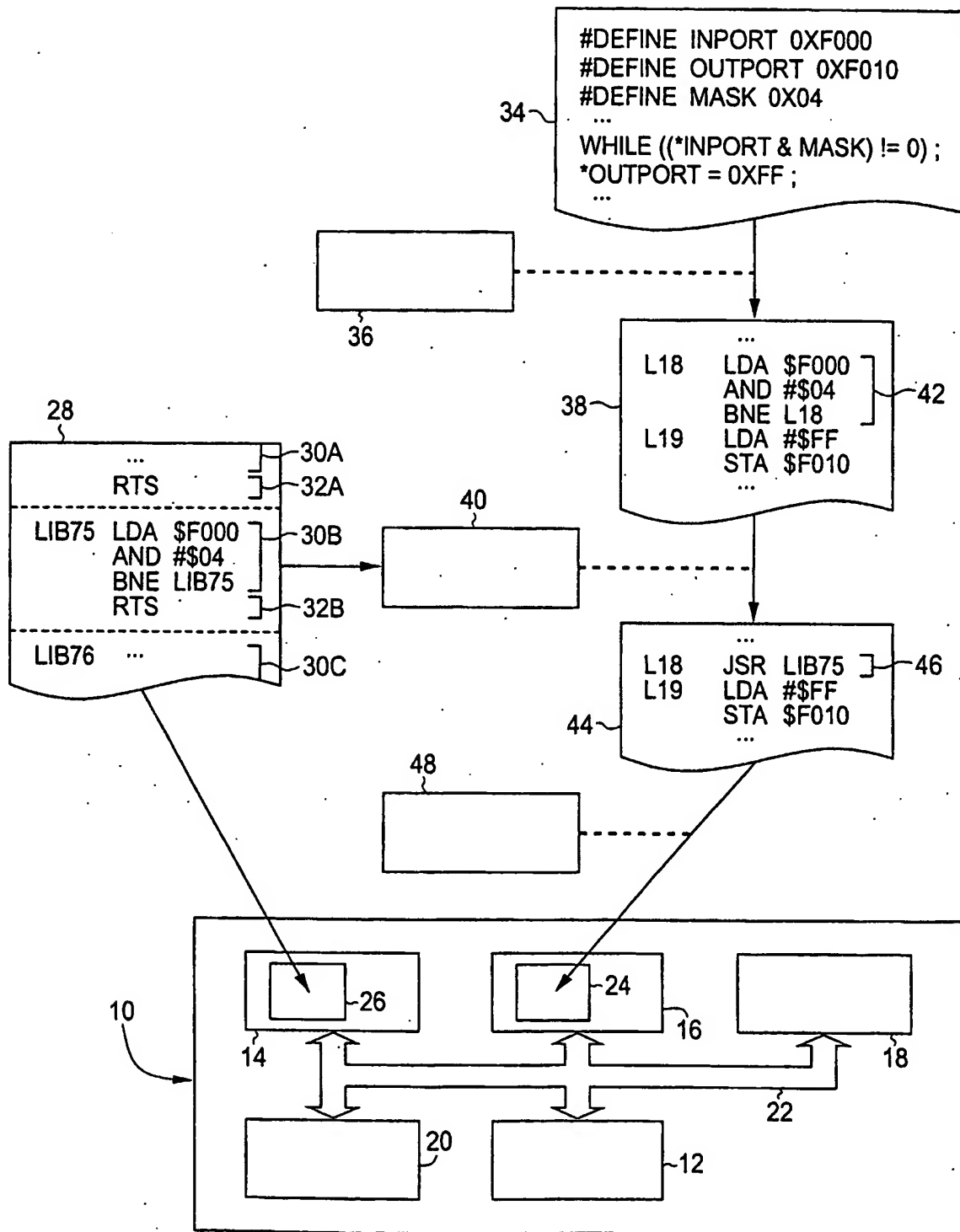


Fig. 1

(12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES  
PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum  
Internationales Büro



(43) Internationales Veröffentlichungsdatum  
23. Oktober 2003 (23.10.2003)

PCT

(10) Internationale Veröffentlichungsnummer  
**WO 2003/088039 A3**

(51) Internationale Patentklassifikation<sup>7</sup>: **G06F 9/45**

(21) Internationales Aktenzeichen: **PCT/EP2003/003803**

(22) Internationales Anmeldedatum:  
11. April 2003 (11.04.2003)

(25) Einreichungssprache: **Deutsch**

(26) Veröffentlichungssprache: **Deutsch**

(30) Angaben zur Priorität:  
102 16 602.1 15. April 2002 (15.04.2002) **DE**

(71) Anmelder (für alle Bestimmungsstaaten mit Ausnahme von US): **GIESECKE & DEVRIENT GMBH** [DE/DE]; Prinzregentenstrasse 159, 81677 München (DE).

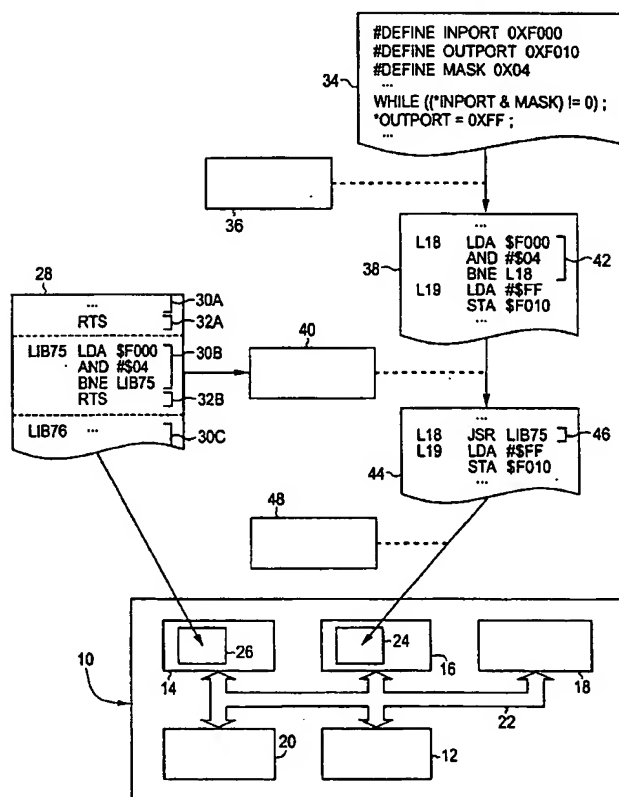
(72) Erfinder; und  
(75) Erfinder/Anmelder (nur für US): **BALDISCHWEILER, Michael** [DE/DE]; Hans-Jakob-Str. 99, 81825 München (DE). **NESS, Werner** [DE/DE]; Ringhofferstr. 76, 85716 Unterschleißheim (DE).

(74) Anwalt: **DENDORFER, Claus**; Wächtershäuser & Hartz, European Patent and Trademark Attorneys, Weinstrasse 8, 80333 München (DE).

[Fortsetzung auf der nächsten Seite]

(54) Title: **OPTIMISATION OF A COMPILER GENERATED PROGRAM CODE**

(54) Bezeichnung: **OPTIMIERUNG VON COMPILERGENERIERTEM PROGRAMMCODE**



(57) Abstract: The invention relates to a method for the optimisation of a compiler generated program code (38). The program code (38) generated by the compiler is examined by program code fragments (42) which at least correspond in relation to the action thereof to a library-code fragment (30x) which is contained in a predefined library (28). The thus discovered program-code-fragments (42) are replaced by respectively calling up the corresponding library-code fragment (30x). A computer program product comprises program instructions for carrying out said method. A mobile data carrier (10) contains the program code (44), which is optimised according to said method, and containing a library. The invention enables efficient utilisation of the memory available in the mobile data carrier (10) and offers a high degree of flexibility for program modifications and/or for the production of data carriers (10) in relatively small quantities.

(57) Zusammenfassung: Bei einem Verfahren zur Optimierung von compilergeneriertem Programmcode 38 wird der compilergenerierte Programmcode 38 nach Programm-Codefragmenten 42 durchsucht, die zumindest hinsichtlich ihrer Wirkung je einem in einer vordefinierten Bibliothek 28 enthaltenen Bibliotheks-Codefragment 30x entsprechen. Die dabei aufgefundenen Programm-Code-Fragmente 42 werden durch je einen Aufruf des entsprechenden Bibliotheks-Codefragments 30x ersetzt. Ein Computerprogrammprodukt weist Programminstruktionen zur Ausführung dieses Verfahrens auf.

Ein tragbarer Datenträger 10 enthält den gemäss diesem Verfahren optimierten

[Fortsetzung auf der nächsten Seite]

WO 2003/088039 A3



(81) **Bestimmungsstaaten (national):** AE, AG, AL, AM, AT (Gebrauchsmuster), AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ (Gebrauchsmuster), CZ, DK (Gebrauchsmuster), DK, DM, DZ, EC, EE (Gebrauchsmuster), EE, ES, FI (Gebrauchsmuster), FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK (Gebrauchsmuster), SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) **Bestimmungsstaaten (regional):** ARIPO Patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), eurasisches Patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), europäisches Patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI Patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

**Erklärung gemäß Regel 4.17:**

- hinsichtlich der Berechtigung des Anmelders, ein Patent zu beantragen und zu erhalten (Regel 4.17 Ziffer ii) für die folgenden Bestimmungsstaaten AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM,

HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW, ARIPO Patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), eurasisches Patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), europäisches Patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI Patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG)

**Veröffentlicht:**

- mit internationalem Recherchenbericht
- vor Ablauf der für Änderungen der Ansprüche geltenden Frist; Veröffentlichung wird wiederholt, falls Änderungen eintreffen

**(88) Veröffentlichungsdatum des internationalen**

**Recherchenberichts:**

4. November 2004

Zur Erklärung der Zweibuchstaben-Codes und der anderen Abkürzungen wird auf die Erklärungen ("Guidance Notes on Codes and Abbreviations") am Anfang jeder regulären Ausgabe der PCT-Gazette verwiesen.

# INTERNATIONAL SEARCH REPORT

International Application No

PCT/EP 03/03803

**A. CLASSIFICATION OF SUBJECT MATTER**  
IPC 7 G06F9/45

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)  
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	FR 2 785 695 A (BULL CP8) 12 May 2000 (2000-05-12) page 6, line 23 - page 12, line 6; figures 2b,4	1-10
A	FR 2 783 065 A (RUE CARTES ET SYSTEMES DE) 10 March 2000 (2000-03-10) page 3, line 20 - page 5, line 20	1-10
A	US 6 263 429 B1 (SISKA CHARLES P) 17 July 2001 (2001-07-17) column 7, line 19 - column 11, line 22	1-10
A	ZASTRE: "Compacting Object Code via Parameterized Procedural Abstraction" THESIS, 1993, XP001199580 UNIVERSITY OF VICTORIA page 28 - page 40	1-10
-/-		

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

\* Special categories of cited documents:

- \*A\* document defining the general state of the art which is not considered to be of particular relevance
- \*E\* earlier document but published on or after the international filing date
- \*L\* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- \*O\* document referring to an oral disclosure, use, exhibition or other means
- \*P\* document published prior to the international filing date but later than the priority date claimed

- \*T\* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- \*X\* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- \*Y\* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- \*G\* document member of the same patent family

Date of the actual completion of the international search

31 August 2004

Date of mailing of the international search report

09/09/2004

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
Fax: (+31-70) 340-3016

Authorized officer

Bijn, K

# INTERNATIONAL SEARCH REPORT

International Application No

PCT/EP 03/03803

## C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 903 759 A (SUN TSYR-SHYA JOE ET AL) 11 May 1999 (1999-05-11) column 6, line 31 - line 47; figure 3 -----	1-10

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/EP 03/03803

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
FR 2785695	A	12-05-2000	FR 2785695 A1	12-05-2000
			BR 9906784 A	17-10-2000
			CA 2317950 A1	18-05-2000
			CN 1292114 T	18-04-2001
			EP 1046105 A1	25-10-2000
			WO 0028416 A1	18-05-2000
			JP 2002529849 T	10-09-2002
FR 2783065	A	10-03-2000	FR 2783065 A1	10-03-2000
US 6263429	B1	17-07-2001	WO 0019309 A2	06-04-2000
US 5903759	A	11-05-1999	NONE	

# INTERNATIONALER RECHERCHENBERICHT

Internationales Aktenzeichen

PCT/EP 03/03803

## A. KLASSIFIZIERUNG DES ANMELDUNGSGEGENSTANDES

IPK 7 G06F9/45

Nach der Internationalen Patentklassifikation (IPK) oder nach der nationalen Klassifikation und der IPK

## B. RECHERCHIERTE GEBIETE

Recherchierte Mindestprüfstoff (Klassifikationssystem und Klassifikationssymbole)

IPK 7 G06F

Recherchierte aber nicht zum Mindestprüfstoff gehörende Veröffentlichungen, soweit diese unter die recherchierten Gebiete fallen

Während der internationalen Recherche konsultierte elektronische Datenbank (Name der Datenbank und evtl. verwendete Suchbegriffe)

EPO-Internal

## C. ALS WESENTLICH ANGESEHENE UNTERLAGEN

Kategorie*	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
A	FR 2 785 695 A (BULL CP8) 12. Mai 2000 (2000-05-12) Seite 6, Zeile 23 - Seite 12, Zeile 6; Abbildungen 2b,4	1-10
A	FR 2 783 065 A (RUE CARTES ET SYSTEMES DE) 10. März 2000 (2000-03-10) Seite 3, Zeile 20 - Seite 5, Zeile 20	1-10
A	US 6 263 429 B1 (SISKA CHARLES P) 17. Juli 2001 (2001-07-17) Spalte 7, Zeile 19 - Spalte 11, Zeile 22	1-10
A	ZASTRE: "Compacting Object Code via Parameterized Procedural Abstraction" THESIS, 1993; XP001199580 UNIVERSITY OF VICTORIA Seite 28 - Seite 40	1-10
-/-		

☒ Weitere Veröffentlichungen sind der Fortsetzung von Feld C zu entnehmen

☒ Siehe Anhang Patentfamilie

\* Besondere Kategorien von angegebenen Veröffentlichungen :

\*A\* Veröffentlichung, die den allgemeinen Stand der Technik definiert, aber nicht als besonders bedeutsam anzusehen ist

\*E\* älteres Dokument, das jedoch erst am oder nach dem internationalen Anmeldedatum veröffentlicht worden ist

\*L\* Veröffentlichung, die geeignet ist, einen Prioritätsanspruch zweifelhaft erscheinen zu lassen, oder durch die das Veröffentlichungsdatum einer anderen im Recherchenbericht genannten Veröffentlichung belegt werden soll oder die aus einem anderen besonderen Grund angegeben ist (wie ausgeführt)

\*O\* Veröffentlichung, die sich auf eine mündliche Offenbarung, eine Benutzung, eine Ausstellung oder andere Maßnahmen bezieht

\*P\* Veröffentlichung, die vor dem internationalen Anmeldedatum, aber nach dem beanspruchten Prioritätsdatum veröffentlicht worden ist

\*T\* Spätere Veröffentlichung, die nach dem internationalen Anmeldedatum oder dem Prioritätsdatum veröffentlicht worden ist und mit der Anmeldung nicht kollidiert, sondern nur zum Verständnis des der Erfindung zugrundeliegenden Prinzips oder der ihr zugrundeliegenden Theorie angegeben ist

\*X\* Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann allein aufgrund dieser Veröffentlichung nicht als neu oder auf erfinderscher Tätigkeit beruhend betrachtet werden

\*Y\* Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann nicht als auf erfinderscher Tätigkeit beruhend betrachtet werden, wenn die Veröffentlichung mit einer oder mehreren anderen Veröffentlichungen dieser Kategorie in Verbindung gebracht wird und diese Verbindung für einen Fachmann naheliegend ist

\*G\* Veröffentlichung, die Mitglied derselben Patentfamilie ist

Datum des Abschlusses der internationalen Recherche

31. August 2004

Absendedatum des internationalen Recherchenberichts

09/09/2004

Name und Postanschrift der internationalen Recherchenbehörde  
Europäisches Patentamt, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
Fax (+31-70) 340-3016

Bevollmächtigter Bediensteter

Bijn, K

# INTERNATIONALER RECHERCHENBERICHT

Internationales Aktenzeichen

PCT/EP 03/03803

C.(Fortsetzung) ALS WESENTLICH ANGESEHENE UNTERLAGEN

Kategorie*	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
A	<p>US 5 903 759 A (SUN TSUR-SHYA JOE ET AL)</p> <p>11. Mai 1999 (1999-05-11)</p> <p>Spalte 6, Zeile 31 - Zeile 47; Abbildung 3</p> <p>-----</p>	1-10

BEST AVAILABLE COPY



# INTERNATIONALER RECHERCHENBERICHT

Angaben zu Veröffentlichungen, die zur selben Patentfamilie gehören

Internationales Aktenzeichen

PCT/EP 03/03803

Im Recherchenbericht angeführtes Patentdokument		Datum der Veröffentlichung	Mitglied(er) der Patentfamilie		Datum der Veröffentlichung
FR 2785695	A	12-05-2000	FR	2785695 A1	12-05-2000
			BR	9906784 A	17-10-2000
			CA	2317950 A1	18-05-2000
			CN	1292114 T	18-04-2001
			EP	1046105 A1	25-10-2000
			WO	0028416 A1	18-05-2000
			JP	2002529849 T	10-09-2002
FR 2783065	A	10-03-2000	FR	2783065 A1	10-03-2000
US 6263429	B1	17-07-2001	WO	0019309 A2	06-04-2000
US 5903759	A	11-05-1999	KEINE		

BEST AVAILABLE COPY